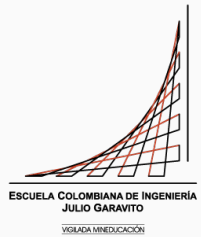

Advanced Natural Language Processing Techniques to Profile Cybercriminals

Proyecto de grado

Autor Alejandro Anzola Ávila
Director Daniel Orlando Díaz López, *PhD*

22 de mayo de 2019

Escuela Colombiana de Ingeniería Julio Garavito

Agenda

1. Objetivos y justificación
2. Resultados propuestos y productos obtenidos
3. Marco teórico
    - Clasificador Naïve Bayes
    - Clasificación con Support Vector Machines (SVM)
    - Self-organizing Maps (SOM)
4. Problemas y soluciones
    - Modelo 1: Predicción de etiquetas de Twitter
    - Modelo 2: Reconocimiento de Named Entities con redes LSTM
5. Conclusiones y trabajo futuro

Objetivos y justificación

Objetivo general

Generar herramientas y estrategias para el perfilado de cibercriminales con ayuda de metodologías de *NLP* aplicado a datos recolectados de comunicaciones y redes sociales.

Objetivos específicos

- Diseñar e implementar una solución de lenguaje natural para realizar el perfilado de sospechosos.
- Identificar el estado del arte en sistemas que usan *NLP* para apoyar agencias de seguridad del Estado.
- Implementación de artefactos para la construcción de *datasets* con información recolectada de medios privados como de fuentes abiertas.
- Validar la solución desarrollada frente a un escenario real.
- Modelado de diferentes metodologías, heurísticas y meta–heurísticas para *NLP*.

Justificación

Por hacer

Resultados propuestos y productos obtenidos

Resultados propuestos

Por hacer

Productos obtenidos i

1. Entendimiento de las generalidades de Data Science:
   - Tipos de Machine Learning
   - Sistemas de detección de anomalías
   - Diferentes modalidades de clustering
2. Identificacion de modelos de NLP aplicables para el perfilado de cibercriminales
3. Entendimiento de los modelos de clasificación y clustering:
   - Clasificador de Naïve Bayes
   - Maquinas de soporte vectorial
   - Mapas autoorganizados
4. Entendimiento de los modelos utilizados en NLP:
   - Predicción de etiquetas con modelos de regresión lineal
   - Reconocimiento de Named Entities

Productos obtenidos ii

   - Uso de *embeddings* generados con StarSpace para los k textos mas similares

5. Implementación de modelos de NLP para el perfilado de cibercriminales:
   - Modelo de predicción de hashtags de Twitter con modelos lineales
   - Modelo de reconocimiento de Named Entities con redes LSTM
   - Modelo de clustering en redes SOM con *embeddings* de StarSpace

Marco teórico

Teorema de Bayes

Para variables aleatorias x e y, se tiene que la probabilidad condicional $P(\text{y} \mid \text{x})$ es definida como

$$P(\text{y} \mid \text{x}) = \frac{P(\text{x} \mid \text{y})P(\text{y})}{P(\text{x})}$$

Clasificador Naïve Bayes

Un clasificador de Naïve Bayes estima la probabilidad condicional de las clases por medio de suponer que los atributos son condicionalmente independientes, dado la etiqueta de clasificación y. Donde cada conjunto de d atributos $\mathbb{X} = \{x_1, \ldots, x_d\}$ se tiene

$$P(\mathbb{X} \mid \mathrm{y} = y) = \prod_{i=1}^{d} P(x_i \mid \mathrm{y} = y)$$

El clasificador computa la probabilidad posterior para cada clase y como

$$P(\mathrm{y} \mid \mathbb{X}) = \frac{P(\mathrm{y}) \prod_{i=1}^{d} P(x_i \mid \mathrm{y})}{P(\mathbb{X})} \Rightarrow P(\mathrm{y}) \prod_{i=1}^{d} P(x_i \mid \mathrm{y})$$

Nota Puede ignorarse $P(\mathbb{X})$ debido a que es un termino constante. Para esto se realiza una normalización con una constante ϵ de forma que $\sum_{\forall \mathrm{y} \in \mathbb{Y}} \epsilon^{-1} P(\mathrm{y} \mid \mathbb{X}) = 1$.

Clasificación con Support Vector Machines (SVM)

Técnica de **clasificación** con una frontera de decisión en forma de hiper-planos que permiten aplicaciones con vectores de alta dimensionalidad.

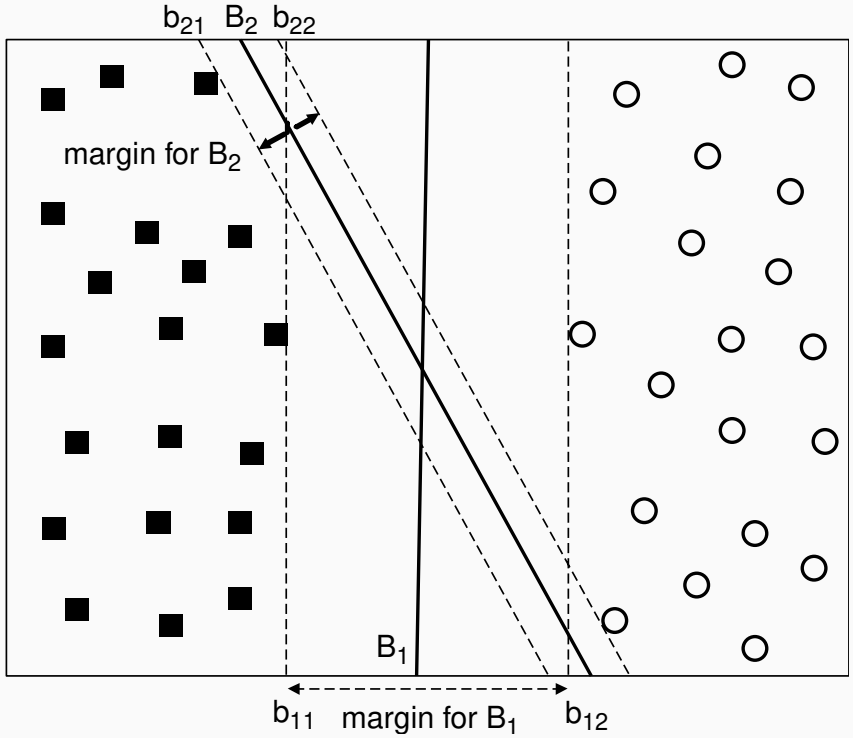

Figura 1: Maximum Margin Hyperplanes. Tomado de [3].

Self-organizing Maps (SOM)

Es un mapa discreto de o neuronas con vectores $\boldsymbol{w} \in \mathbb{R}^m$ que se adaptan a una entrada de $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ de N patrones. Tiene una adaptación con una tasa de aprendizaje α_t y un área de afectación σ_t que se reducen por cada iteración $t \in \{0, \ldots, T\}$.

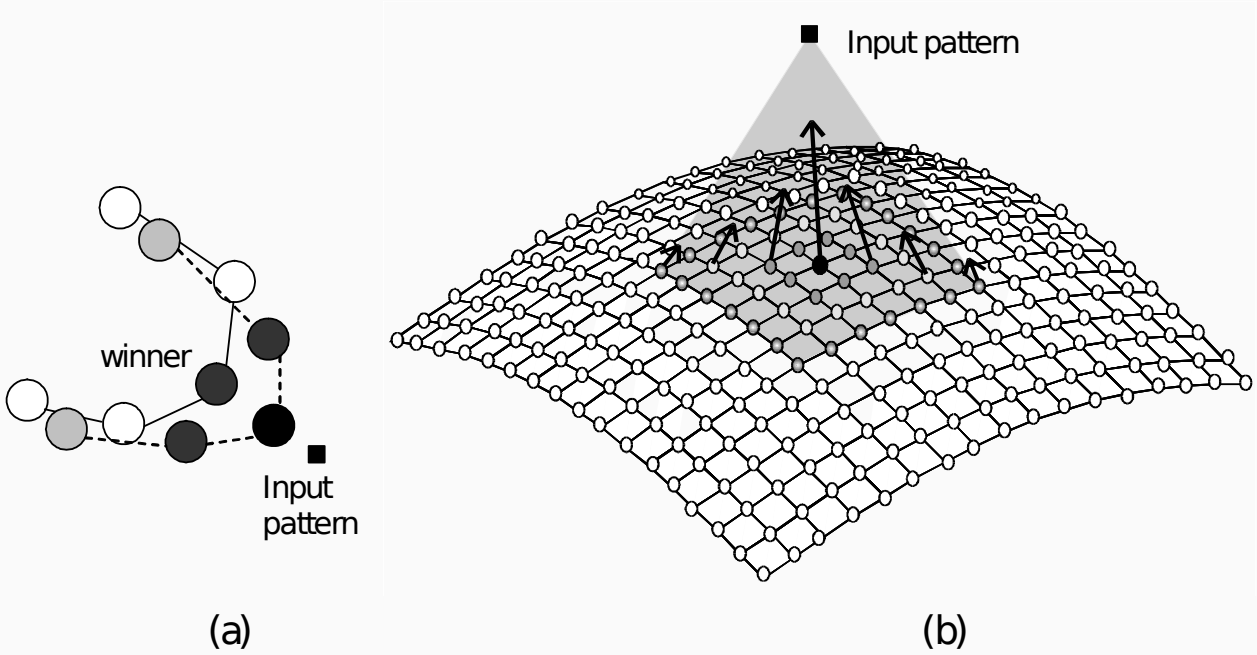

Figura 2: Proceso de adaptación de SOM, (a) uni–dimensional, (b) bi–dimensional. Tomado de [1].

Ejemplo de SOM

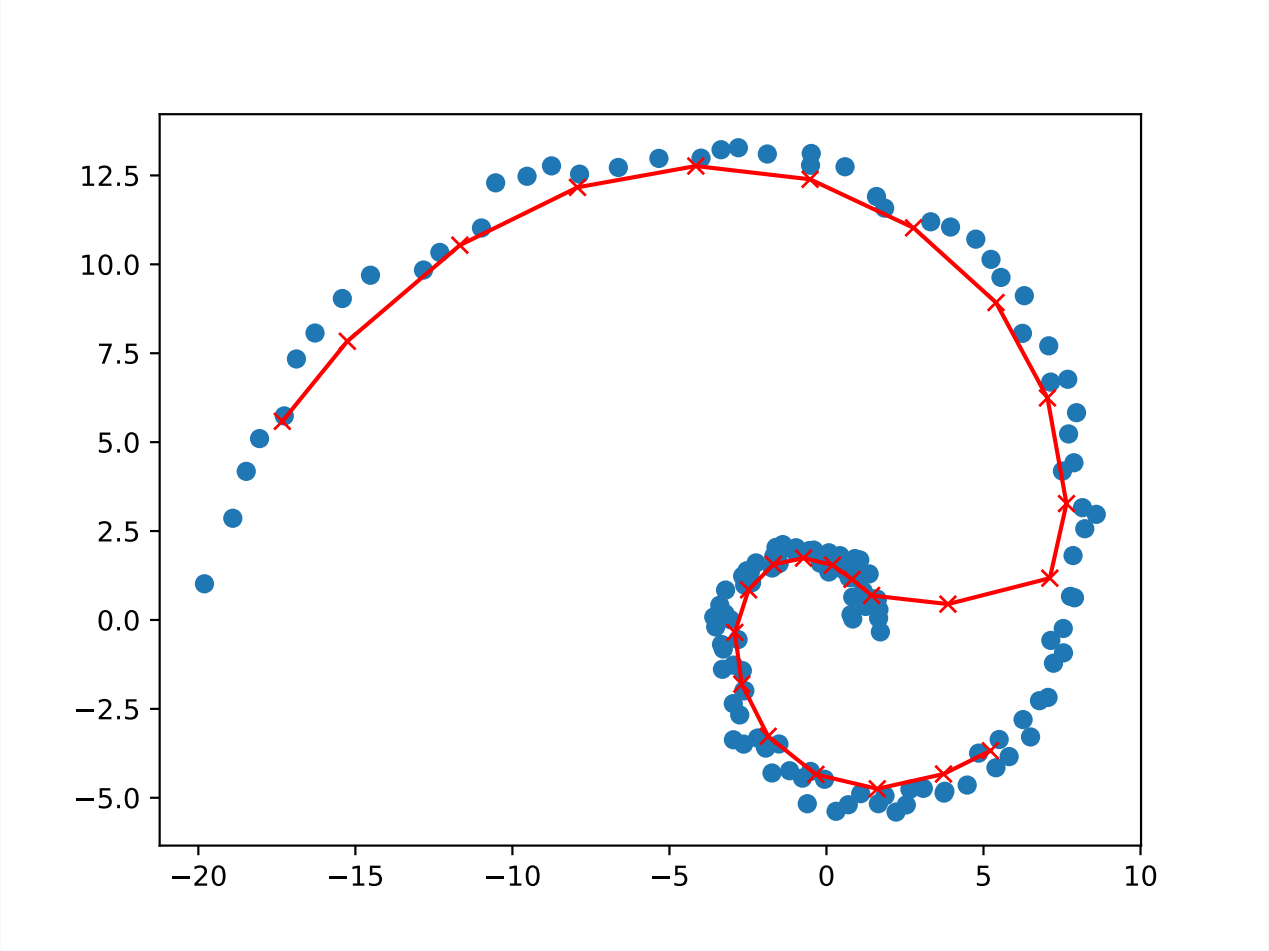

Figura 3: Ejemplo de salida de SOM uni-dimensional con 25 neuronas. Implementación propia.

Aplicación de SOM en perfilamiento de criminales

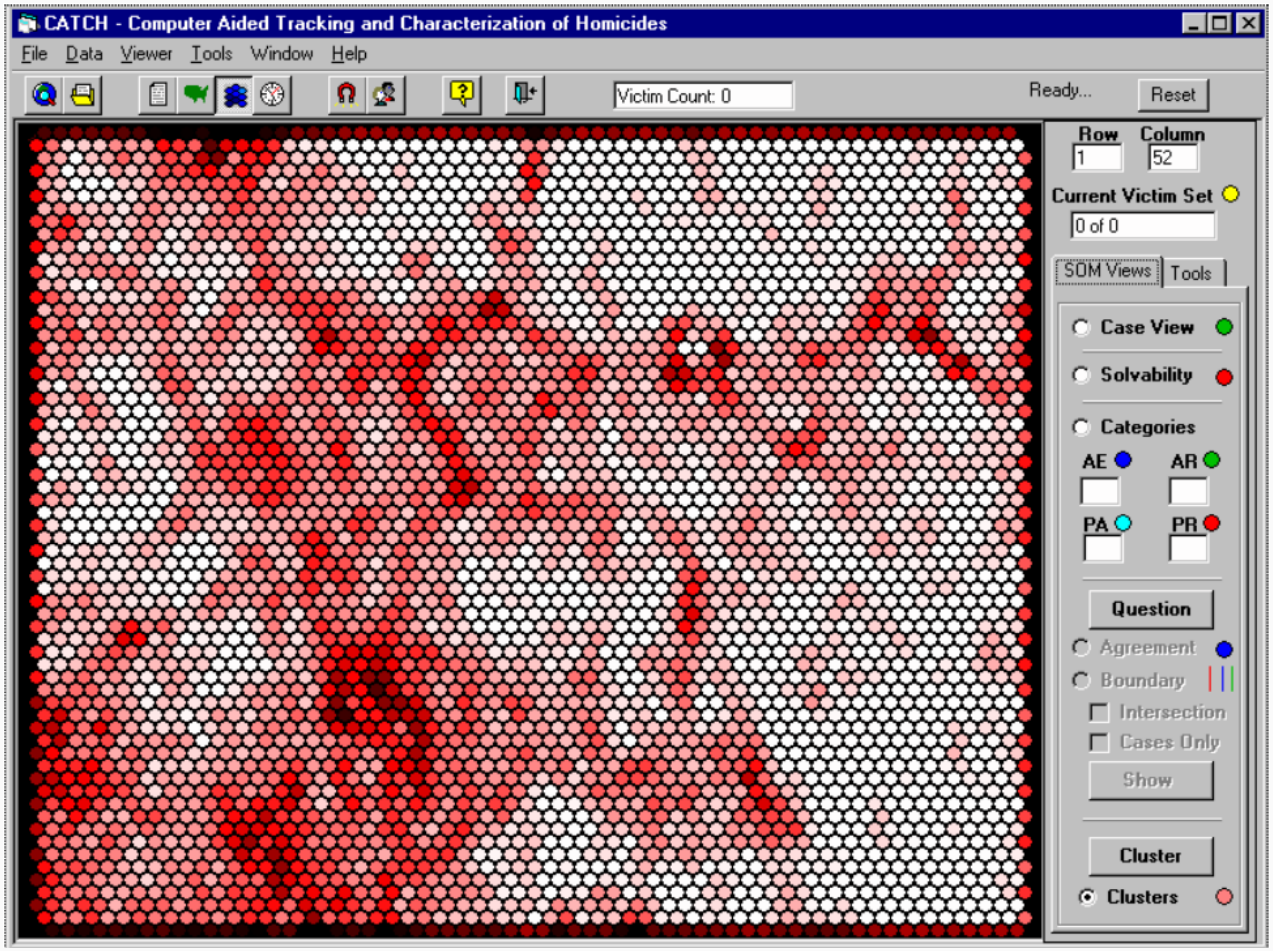

Figura 4: Ejemplo de uso de SOM en aplicaciones de perfilado. Tomado de [2].

Problemas y soluciones

Modelo 1: Problema

Tweet

> *"Really excited to add @plaidavenger to my* ***#deathlist*** *along with Italy and @Plaid_Obama after receiving that information.* ***#KillEveryone #ISIS****"*

Modelo 1: Solución

¿Que hacer?
Con un modelo de regresión lineal predecir los hashtags de los tweets.

"Really excited to add @plaida-venger to my deathlist along with Italy and @Plaid_Obama after receiving that information." $\Rightarrow$ **#deathlist, #KillEveryone, #ISIS**

Representación de palabras: Bag of Words

N es el tamaño del diccionario de términos D (e.g. $N = |D|$).

$$\text{word2idx} = \Big\{(t_i, i) : \forall i \in \{1, \ldots, N\}\Big\}$$

$$\text{idx2word} = \Big[t_1, \ldots, t_N\Big]$$

Representación de palabras en vectores para BoW
Para un termino individual su vector representativo se define como:

$$\boldsymbol{e}^{(i)} = [0, \ldots, 1, \ldots, 0] \leftarrow \text{posicion } i\text{–ésima}$$

$$\boldsymbol{e}^{(i)}, (t, i) \in \text{word2idx}$$

Para un documento d de términos, se calcula por cada termino que existen dentro del diccionario su vector representativo como:

$$\boldsymbol{s} = \sum_{(t,i) \in \text{word2idx}} \boldsymbol{e}^{(i)}, t \in d$$

Representación de palabras: TF–IDF

TF–IDF = Term Frequency – Inverse Document Frequency

Proposito
Darle mayor importancia a las palabras que ocurren con frecuencia intermedia en el documento d y en el corpus D.

$$\text{tf}(t, d) = \text{Frecuencia del termino (o n–grama) } t \text{ en el documento } d$$

$$\text{idf}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right); N = |D|$$

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

Regresión lineal

Para una vector de parámetros $\boldsymbol{\theta}$ y un vector de características $\boldsymbol{x}$, la regresión lineal se puede definir como:

$$\hat{y}(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x} = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

Donde $\hat{y}(\boldsymbol{x}, \boldsymbol{\theta}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.

θ_0 se le conoce como el *bias* del modelo.

El objetivo es que para una salida esperada y se tenga la salida $\hat{y}$ con menor error por medio de ajustar los valores de $\boldsymbol{\theta}$. De forma que se quiere:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} |\hat{y}(\boldsymbol{x}, \boldsymbol{\theta}) - y|$$

Regresión logística $\sigma(x)$

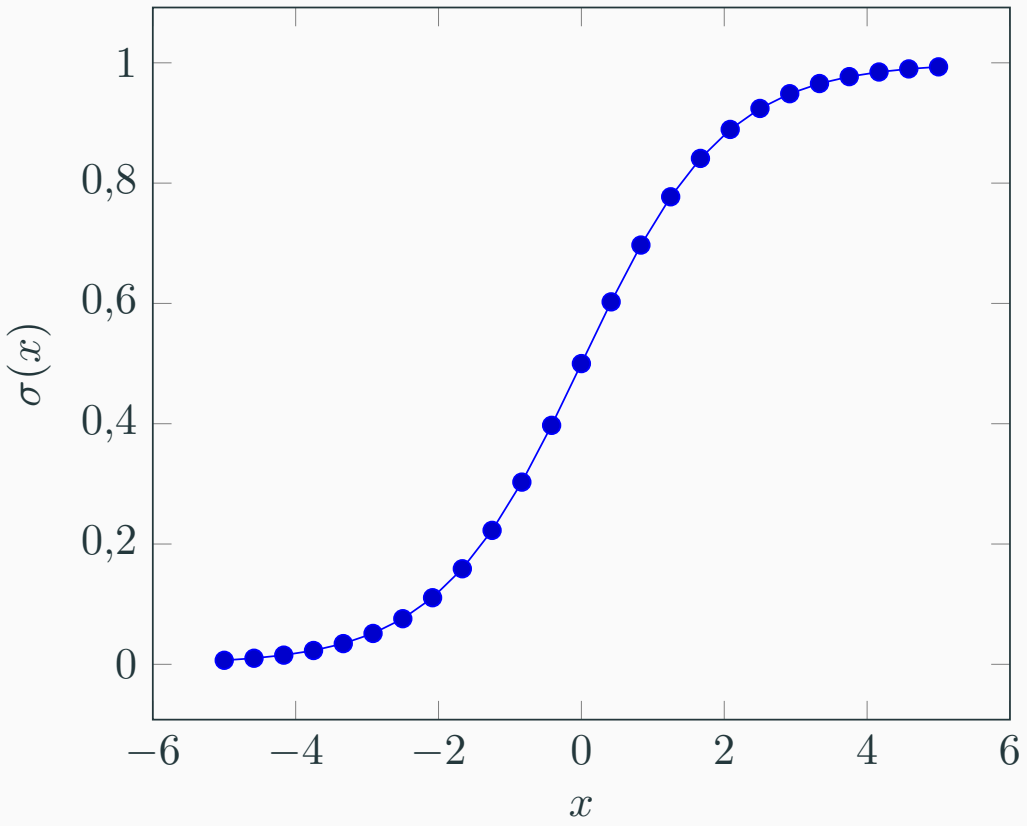

Figura 5: Gráfica de función sigmoide.

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

$$\sigma(x) : \mathbb{R} \to (0, 1)$$

Evita problemas de BIAS y OVERFITTING del modelo

One vs Rest

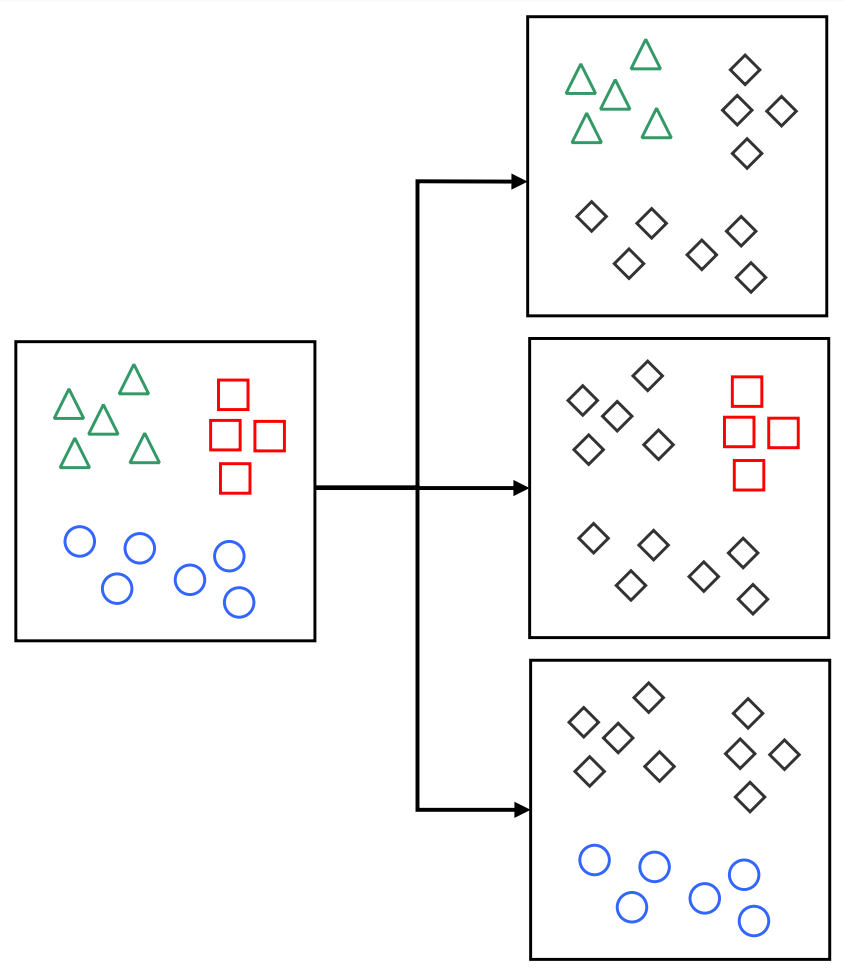

Figura 6: Algoritmo de One vs Rest.

Se entrenan C estimadores para cada clase con algún algoritmo de optimización (ej. gradiente descendiente).

Se determina un estimador $c \in \{1, C\}$, que se calcula como:

$$c = \arg\max_i \sigma(\boldsymbol{\theta}_i^\top \boldsymbol{x})$$

Modelo 2: Reconocimiento de Named Entities con redes LSTM

Por hacer

Conclusiones y trabajo futuro

Conclusiones

Por hacer

Trabajo futuro

Por hacer

Bibliografía i

[1] L. N. De Castro.
Fundamentals of natural computing: basic concepts, algorithms, and applications.
Chapman and Hall/CRC, 2006.

[2] J. Mena.
Investigative Data Mining for Security and Criminal Detection.
Elsevier Science, 2003.

[3] P.-N. Tan, M. Steinbach, and V. Kumar.
Introduction to Data Mining.
Addison Wesley, us ed edition, May 2005.